



POLITECNICO DI TORINO

DIGITAL SYSTEMS ELECTRONICS  
A.A. 2018/2019

PROF. G. MASERA

---

## Lab 10

11 june 2019

---

Berchialla Luca	236032
Laurasi Gjergji	238259
Mattei Andrea	233755
Lombardo Domenico Maria	233959
Wylezek Karolina	267219

# 1 Measuring the frequency and the duty cycle of a squarewave

The first task of the laboratory consists in measuring the frequency of a squarewave using the input capture function of timer 3 (TIM3). The input squarewave is received on the pin PA6. The only code present in "main.c" file (aside from the one generated by CubeMX) is the one used for enabling timer in input capture mode and the relative interrupt. In the TIM3 interrupt function, after checking if the interrupt has been called by the given input capture, the value of the time stamp of the rising or falling edge is saved in a variable. After the second rising edge the period is calculated as the difference between the 2 rising edge stamps by the prescaler and divided by the clock frequency, similarly the duty cycle is measured by capturing the falling edge event. In this way, both positive and negative semi period can be independently measured.

The prescaler has been set in order to have a 42kHz frequency (prescaler:1999), therefore in case of 100Hz waves the timer cannot reset multiple time.

If we put the break-point where we haven't already acquired all the count0, count1, count2 we can't get the right value because even though the execution of the code is stopped, the counter continue counting.

The maximum detectable frequency is equal to the counter's clock frequency because we have at least to count to 1, so to increase it we can lower the prescaler value. The minimum detectable frequency in our code is equal to the counter's clock times  $2^{16}$  which is the value of the ARR register. Actually we can further increase the maximum detectable period if we keep track of the ARR flag.

To increase the resolution of the measurement of a single period we have to increase the clock frequency by lowering the prescaler value.

## 2.1 Generating a square wave

The code for this task is very similar to the one of the previous part. A square wave with frequency 100Hz and duty cycle ranging between 25% and 75% has been generated using the PWM function of the timer 3 on pin PA6. In order to have the desired frequency the prescaler was set to 10 and the auto-reload was set to 42. The changing DC is set up by reading the frequency of a sample input square wave.

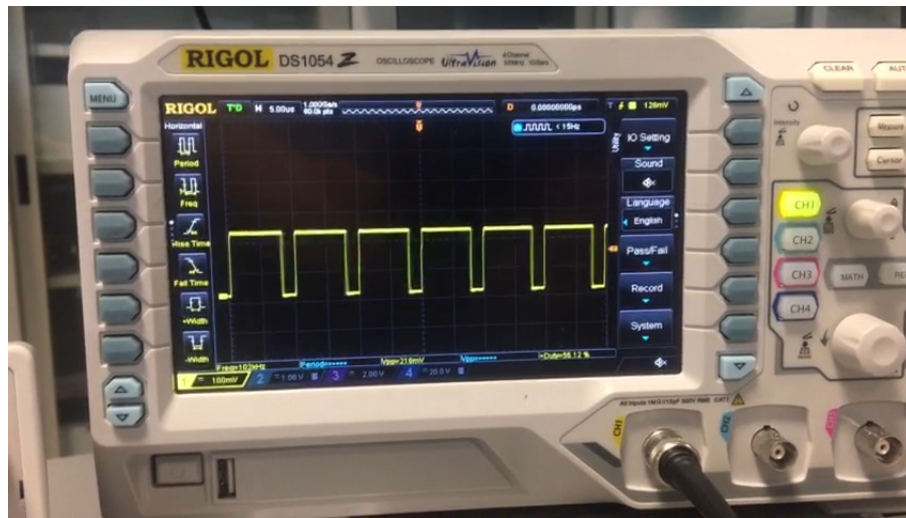


Figure 1: DC=75%



Figure 2: DC=25%

## 2.2 LED dimmer

The main difference of this part from the previous ones is the fact that we shift from the LL libraries to the HAL ones. After starting the PWM generation (with the `HAL_TIM_PWM_Start` function), the duty cycle is continuously increased up to 100% and then decreased down to 0 with the `__HAL_TIM_SET_COMPARE` function. The given delay specifications have been satisfied exploiting the HAL function `HAL_Delay`.

If we double the period and halve the prescaler we still see a square wave at 100kHz, but the DC varies between 0 and 50% because now the period is doubled and but we didn't double the value at which we turn the output off.

Since the max DC is 50% the led results less bright than before, but nothing else has changed.

## 2.3 Generation of a sine waveform by means of PWM

The final task required to build a sine wave generator using the PWM function of the microcontroller. The wave is generated from 200 samples of sine used for changing the duty cycle of the PWM dynamically. In order to achieve high performances the DMA mode of the PWM is used for updating the value of the CCR register. The output signal (generated by TIM3 on the PA6 pin) needs to be filtered using an hardware low-pass filter composed of a  $220\Omega$  resistance and a  $1\mu\text{F}$  capacitor.



Figure 3: Filtered sine wave and unfiltered PWM signal